Written by Larry Ashmun

Disk Modifier for Hobbiest or Businessmen

For Mod III Only

Written for DosPlus & TRS-DOS

Distributed by:

SSM SOFT SECTOR MARKETING, INCORPORATED
6250 Middlebelt • Garden City, Michigan 48135
(313) 425-4020

Zapsit

written by Larry Ashmun
for the TRS80 MODEL III

Zapsit is a stand alone machine language program that lets you examine, modify, copy disk sectors and much more. It does not use any of the resident DOS routines so that you are not limited to the restrictions of the particular DOS that you normally use. You do not have to have a system disk in DRIVE 0 once Zapsit is running.

Currently there are two versions of Zapsit - one for use with *TRSDOS and one for use with **DOSPLUS 3.3 or on DOSPLUS 3.4 single sided disks. They are the same except in the way that they format a disk and the way that they write DATA ADDRESS MARKS. Because of the differences, they are not interchangable. Writing to a TRSDOS diskette with the DOSPLUS version (or vice-versa) could make the diskette written to unuseable.

Both versions come on the same diskette with a loader that lets you select the version that you want to use.

** To load programs put in drive 0 and press RESET button. **

DOSPLUS Zapsit loads from 7000H to 9020H and starts execution at 7242H.
TRSDOS Zapsit loads from 7000H to 8FFFH and starts execution at 7216H.

Either version can be loaded into memory and dumped to a CMD file on a system diskette.

* TRSDOS is a trademark of TANDY Corporation
** DOSPLUS is a trademark of MICRO SYSTEMS SOFTWARE, INC.

WARNING !!

Before attempting to use this program to modify or copy diskettes read ALL OF THE DOCUMENTATION !!!

This is a very powerful and versatile program. You should make every attempt to understand how this program works and exactly what you want to do before using it on any of your important diskettes. Only work on backups whenever possible. MISTAKES CAN BE HAZARDOUS TO THE HEALTH OF YOUR DATA.

The author, Soft Sector Marketing Inc., nor any of the distributers can accept any responsibility or liability for any damages or losses caused or alleged to have been caused by the use of this program.

GENERAL INFORMATION

For all disk operations you will be asked a series of questions. Each question can be answered by pressing the ENTER key or by entering a specific value. Pressing the ENTER key will cause the DEFAULT value to be used. All default values are indicated on the screen at the time the question is asked.

When entering a specific numeric value it is assumed to be DECIMAL unless an H is appended to the number. When an H is appended, the number is assumed to be hexadecimal (base 16). eg. 10 (decimal) = AH (hexadecimal). For a further discussion on different numbering systems used with computers read the Z80 HANDBOOK by WILLIAM BARDEN Jr.

Following are the questions asked for most of the DISK operations in Zapsit:

```
Drive # (0-3) ?  Default = 0
```

Answer this question with the number of the drive that you want to work with. At this point, if you have DOUBLE SIDED disk drives, you can also select which SIDE of the diskette that you want to work with. Typing 1 or 1A will let you work on SIDE A (the normal side) of the diskette in drive 1 - typing 1B will let you work with SIDE B (the back side) of the diskette. The SIDE default is SIDE A.

Pressing ENTER will cause DRIVE 0, SIDE A to be used.

```
D = Double Density  <- Default
S = Single Density              ?
```

Pressing ENTER or typing D will cause Zapsit to use DOUBLE DENSITY reads and writes to the diskette.

Typing S will cause Zapsit to use SINGLE DENSITY reads and writes. With SINGLE DENSITY you can work on MODEL I diskettes (with some restrictions explained in the SET DIRECTORY TRACKS section).

```
Track #   LT to HT   Default = LT     ?
```

LT = the LOWEST TRACK number allowed. Normally 00 for TRSDOS or DOSPLUS. It can be changed by you at Change Track/Sector limits.

HT = the HIGHEST TRACK number allowed. Normally 39 for TRSDOS or DOSPLUS. It can be changed by you at Change Track/Sector limits.

Pressing ENTER will cause the LOWEST TRACK number allowed to be used.

```
Sector #  LS to HS   Default = LS     ?
```

LS = the LOWEST SECTOR number allowed. Normally 00 for DOSPLUS or 01 for TRSDOS. It can be changed by you at Change Track/Sector limits.

HS = the HIGHEST SECTOR number allowed. Normally 17 for DOSPLUS or 18 for TRSDOS. It can be changed by you at Change Track/Sector limits.

Pressing ENTER will cause the LOWEST SECTOR number allowed to be used.

NOTE : If you are using the SINGLE DENSITY MODE different default and limit values will be displayed but they work the same way and can be changed the same way as the DOUBLE DENSITY MODE (See Change Track/Sector limits).

Some of the disk operations will ask you for additional information which will be discussed in the sections that need it.

```
                Main Menu
                =========

     0 = Display/Modify Disk Sectors      <- Default
     1 = Display/Modify Memory
     2 = Change Track/Sector Limits
     3 = Set Directory Tracks
     4 = Format a Disk
     5 = Verify a Disk
     6 = Copy Disk Sectors
     7 = Zero Disk Sectors
     8 = Read a Track
     9 = Return to DOS

   Your Choice ?
```

To make your selection, type the key that corresponds to what you want to do and press ENTER.

Each of the selections on this menu will be explained in the order that it appears on the menu

Pressing the BREAK key anywhere in Zapsit will abort the operation being performed and return you to this menu.

Display/Modify Disk Sectors

After answering the questions about the disk (drive #, density, track #, sector #) the specified sector will be read into a buffer (temporary storage area located at 7000H) and be displayed in the following format (without the column numbers):

```
 1 2 3                    4                                        5
 = = =  ======================================== ================
000100  FE11 3ED0 D3F0 2102 0022 EA43 AF32 EC43 ..>...!.."C.2.C
000110  CD3E 43FE 0128 0CFE 0220 E7CD 3E43 CD35 .>C..(... ..>C.5
000120  43E9 FFCD 3E43 D602 47CD 3543 CD3E 4377 C...>C..G.5C.>Cw
000130  2310 F918 DBCD 3E43 6FCD 3E43 67C9 C5E5 #.....>Co.>Cg...
000140  3AEC 43B7 202E 0609 C5CD 7F43 C1E6 1D28 :.C. ......C...(
000150  133E D0D3 F010 F13E 17CD 3300 21ED 43CD .>.....>..3.!.C.
000160  1B02 18FE 2AEA 432C 7DFE 1338 032E 0124 ....*.C,...(...$
000170  22EA 43AF 6F26 4D3C 32EC 437E E1C1 C9CD ".C.o&M<2.C.....
000180  C543 01F3 003E 81D3 F457 21B7 4322 4A40 .C...>...W!.C"J@
000190  3EC3 3249 40F3 3EC0 D3E4 1E02 2100 4D3E >.2I@.>.....!.M>
0001A0  84D3 F0CD E043 DBF0 A328 FBED A27A F640 .....C...(...z.@
0001B0  D3F4 EDA2 C3B0 43E1 AFD3 E43E 81D3 F4CD ......C....>....
0001C0  E643 DBF0 C93E 81D3 F42A EA43 7CD3 F33E .C...>...*.C...>
0001D0  1CD3 F0CD E643 DBF0 CB47 20FA 7DD3 F2C9 .....C...G .....
0001E0  F5F1 F5F1 F5F1 F5F1 00C9 0200 0017 4552 ..............ER
0001F0  534F 520D 0110 4062 4810 0253 0000 1328 ROR...@bH..S...(
```

The columns are numbered at the top and will be referenced by their number in the following explanations.

Column 1 is the current TRACK # (hexadecimal) that is being displayed. In the example it is track 00.

Column 2 is the current SECTOR # (hexadecimal) that is being displayed. In the example it is sector 01.

Column 3 is the relative position (hexidecimal) of the first byte of that row within the sector being displayed. If you go down to the row labeled 000160 you will see that the first byte in the row is 1BH. That byte is byte number 60H (96th decimal) within the sector (all counting starts with the number 0).

Column 4 is the actual data (hexadecimal) that is stored in the sector being displayed. Each grouping of 4 digits represents 2 bytes.

Column 5 is the ASCII representation of the numbers in that row. If the number does not coincide with an ASCII character a period (.) is substituted. This substitution can be overridden by pressing the CLEAR key (turning off the graphics filter) and re-reading the sector. When the filter is turned off all control codes and graphics characters will be displayed. The filter can be turned back on again by pressing the CLEAR key again.

When in the Display/Modify Disk Sectors mode there are several commands available to you. The following is an explanation of these commands.

Right Arrow key

Display the NEXT HIGHER sector. If you are on the highest sector of the track, the LOWEST SECTOR of the NEXT HIGHER TRACK will be displayed.

SHIFT, Right Arrow key

Display the HIGHEST SECTOR on the HIGHEST TRACK allowed. The TRACK/SECTOR limits can be set from the main menu selection # 2

Left Arrow key

Display the NEXT LOWER SECTOR. If you are on the lowest sector of the track, the HIGHEST SECTOR of the NEXT LOWER TRACK will be displayed.

SHIFT, Left Arrow key

Display the LOWEST SECTOR of the LOWEST TRACK allowed. The TRACK/SECTOR limits can be set from the main menu selection # 2.

Up Arrow key

Display the SAME SECTOR # on the next LOWER TRACK.

Down Arrow Key

Display the SAME SECTOR # on the next HIGHER TRACK.

T Key

Pressing the T key allows you to enter the Track and Sector numbers to display (same drive).

P key

Pressing the P key sends the current sector displayed to the Line Printer Port in the same format as the screen display. While the printer is printing, pressing the BREAK key will abort the printing and return you to the main menu. If Zapsit detects that a printer is not available the operation will be aborted and you will be returned to the main menu.

CLEAR key

Pressing the CLEAR key turns the Graphics Filter ON or OFF.

Q or BREAK key

Pressing the Q or BREAK key will return you to the main menu.

MODIFY MODES

M key

Pressing the M key will put you in the HEXADECIMAL MODIFY MODE. It allows you to change the numbers in column 4. The four arrow keys will allow you to move around the buffer without changing anything. A flashing graphics block cursor on the screen display will show you your position in the buffer. You can change the numbers by typing the new numbers (hexadecimal) over the top of the old numbers. If you make a mistake and want to cancel all the changes that you have made press SHIFT, DOWN ARROW X (control X) and the changes will be cancelled. The sector will then be read back into the buffer and re-displayed.

Another way to cancel the changes is to press the ENTER key, then press the C key when you are asked if you want to cancel the changes or update the disk.

One last way to cancel is to press the BREAK key. The BREAK key will abort the operation and return you to the main menu.

A key

Pressing the A key puts you into the ASCII MODIFY MODE. It allows you to change the data in column 5 using the ASCII characters instead of looking up the Hex equivalents. The Arrow, Control X, ENTER, and BREAK keys work in the same way as in the HEXADECIMAL MODIFY MODE.

ENTER key

When you are in either modify mode, pressing the ENTER key terminates the mode. You will then be asked if you want to cancel the changes or update the disk. Press C to cancel - U to update. If you press C the changes will be cancelled and the original sector will be re-displayed. If you press U the changes will be written to the disk and then the sector will be re-displayed with the changes.

Display/Modify Memory

When entering this section of Zapsit you will be asked for a starting address. The address can be entered in either hexadecimal (appended H) or decimal. The program will calculate which PAGE OF MEMORY the address falls in and display that page of memory, starting with the FIRST ADDRESS OF THE PAGE. A page

of memory is a 256 byte block of memory starting with a HEX address that ends in two zeros. Address 4F00H is the first address of page 4FH of memory. The page of memory will be displayed in the same format as Display/Modify Disk Sectors (see example on page 4) with the following exceptions.

Column 1 will always be zeros.

Columns 2 & 3 make up the address (hexadecimal) of the first byte of the row.

The commands available to you in this mode are as follows:

Up Arrow key

Display the NEXT LOWER page of memory. If you are on the lowest page the HIGHEST page of memory will be displayed.

Down Arrow key

Display the NEXT Higher page of memory. If you are on the highest page the LOWEST page of memory will be displayed.

P key

Pressing the P key will send the page of memory being displayed on the screen to the printer port. If Zapsit detects that a printer is not available the operation will be aborted and you will be returned to the main menu.

Q or BREAK key

Pressing the Q or BREAK key will return you to the main menu.

M or A key

The M or A key puts you in a modify mode the same as in Display/Modify Disk Sectors.

SHIFT, Down Arrow, X (control X)

Control X and the BREAK key are the only ways to cancel the changes when you are in either the ASCII or HEX memory modify mode. Pressing ENTER will make the changes without asking you if you want to cancel.

CLEAR key

The CLEAR key works the same as in Display/Modify Disk Sectors.

Change Track/Sector Limits

When you enter this section of Zapsit you will be asked to enter the track & sector numbers in the following order:

Double Density

```
Highest Track #  -- Currently xx     ?
Lowest Track #   -- Currently xx     ?
Highest Sector # -- Currently xx     ?
Lowest Sector #  -- Currently xx     ?
```

Single Density

```
Highest Track #  -- Currently xx     ?
Lowest Track #   -- Currently xx     ?
Highest Sector # -- Currently xx     ?
Lowest Sector #  -- Currently xx     ?
```

xx = The value (decimal) that Zapsit is currently using.

The Track/Sector limits are used for ALL DRIVES in the system.

When answering the questions, pressing ENTER will skip that parameter without changing it.

EXAMPLE: If you only want to change the LOWEST TRACK limit for DOUBLE DENSITY, do the following:

1. Press ENTER when it asks for the highest track limit.

2. Enter the new LOWEST track limit and press ENTER.

3. Press the BREAK key.

That sequence will change only the lowest track limit for double density and return you to the main menu. Other changes or combinations of changes can be made in a similar manner.

Now that you know HOW to change the limits you may be wondering WHY you might want to change them.

Here's WHY - If you have a diskette with one bad track but with valid data on the rest of the tracks you wouldn't want to re-format the whole disk. Say the bad track is track 10 and the disk is double density you could do the following:

1. Set the DOUBLE DENSITY HIGHEST TRACK number and the LOWEST TRACK number to 10.

2. Go back to the main menu and select FORMAT a Disk (selection # 4).

3. Answer the questions for what type of format, drive # and density, and Zapsit will take care of the rest.

When Zapsit is first loaded into memory and run, the HIGHEST TRACK number for double and single density is set to 39. With it set that way you wouldn't be able to read all of an 80 track diskette, if one of your drives happens to have 80 tracks. Changing the HIGHEST TRACK # to 79 would allow you to read/write to all of an 80 track diskette.

ALL DISK I/O ROUTINES HONOR the TRACK LIMITS. That is, they will NOT attempt to do a read or write on a track number higher or lower than the limits set for track numbers.

ALL DISK I/O ROUTINES EXCEPT FORMAT HONOR the SECTOR LIMITS. That is, they will NOT attempt to read or write to a sector number higher or lower than the limits set for sector numbers.

Format will honor the LOWEST SECTOR number as the sector number to start formatting the diskette with but will ignore the HIGHEST SECTOR #.

In double density it will start at the lowest sector number and format 18 sectors with an increment of 1 between sectors. In other words - if the lowest sector is 0 the highest sector will automatically be 17 - if the lowest sector is 1 the highest sector will be 18. Any range of sector numbers can be used as long as the highest sector number does not exceed 255 (decimal). If the highest sector goes above 255 it will wrap around to 0 and continue from there. Does that bring to mind any ideas ? The only problem with wrap-around is that it does strange things to Zapsit's limit checking -- do it at your OWN RISK.

The same holds true for single density diskettes except that only 10 sectors will be formatted.

At the beginning of this documentation I mentioned that the TRSDOS version of Zapsit cannot write to a DOSPLUS diskette and vice-versa - one of the reasons for this is the way the two operating systems number the sectors. TRSDOS numbers the sectors 01 thru 18 - DOSPLUS numbers them 00 thru 17. It would seem that all a person would have to do is change the sector limits and then everything would be ok -- NOT TRUE -- the physical order of the sectors on the disk is also different and the DATA ADDRESS MARKS are handeled differently on the two systems (see the READ A TRACK section).

Even though you cannot WRITE to the wrong type of disk, you CAN read it.

SET DIRECTORY TRACK

When you first enter this section of Zapsit you will be asked for the directory track number for each drive in the following order:

```
Drive 0 -- Currently xx      ?
Drive 1 -- Currently xx      ?
Drive 2 -- Currently xx      ?
Drive 3 -- Currently xx      ?
```

xx = the value that Zapsit is currently using. All drives are set at track # 17 (decimal) when Zapsit is first run.

Pressing ENTER for any of the drives causes that drive to be skipped without changing the parameter.

Setting the DIRECTORY track lets Zapsit know where to expect the directory to be on the diskette. This is not important to Zapsit for reading the disk but is VERY IMPORTANT if you want to write new information to the disk. Especially if you want to change the information on the directory.

When a sector on the directory track is written to the disk it is written with a different DATA ADDRESS MARK (see read track) than the non-directory sectors on the disk. If the correct ADDRESS MARK is not there the DOS cannot verify that it is reading the directory and the results are unpredictable. Some operating systems are not bothered by this but others refuse to even BOOT if the directory has the wrong DATA ADDRESS MARKS. When writing to the disk, play it safe, make sure that the directory track is set to the same track number that the directory is actualy on. This has to be set for EACH DRIVE in your system. Failure to do so could result in the disk being unreadable by the operating system on the disk !

When writing to a single density diskette, the directory track setting does not matter. The disk controller chip (Western Digital 1791 or 1793) used in the MODEL III is not capable of writing the correct DATA ADDRESS MARK to the directory sectors of a MODEL I diskette. If you do modify a directory sector on a MODEL I diskette you will have to READ PROTECT the directory on a MODEL I system using Super Utility or Super Zap. That is an unavoidable problem caused by the differences in the hardware.

FORMAT A DISK

When you select Format you will be asked:

```
S = Standard Format      <- Default
W = Format without Erase

 Which ?
```

If you press ENTER or S you will be asked which drive, what density, and the fill byte to use. The fill byte is the data that is written into the new sectors - ideally, 5BH should be used for double density and E5H for single density - any value below F0H, 240 decimal, will be accepted.

After answering these questions Zapsit will format the disk using the parameters specified. As each track is formatted all of the sectors are checked for readability before the next track is formatted. Any unreadable sectors are reported but the operation will not be aborted.

No checking is done, before formatting, to see if the disk has data and all existing data on the diskette will be lost.

If you type W you will be asked which drive, what density to use and if you want any bad sectors reported to the printer. Before each track is formatted each readable sector on the track is read into a holding area. Unreadable sectors are reported and their holding area is zeroed. After all of the sectors for a track are stored, the track is reformatted and the data in the holding areas written to the appropriate sectors. It is recommended that a DISK VERIFY (selection # 5 on the menu) be performed after a FORMAT without ERASE because no comprehensive verifying is done during this type of formatting (disk write errors will be detected though).

Formatting selected tracks, formatting only one or a few tracks as opposed to formatting the entire disk, can be accomplished by setting the Track/Sector limits to the desired values before formatting.

VERIFY A DISK

Verify a disk does just that. It verifies that every sector within the Track/Sector limits is readable. Any unreadable sectors are reported. You will be asked if you want the unreadable sectors reported to the printer.

If the same sector on all of the tracks is unreadable the disk may have a non-standard format or you may have the Track/Sector limits set incorrectly.

If the same sector/s on several contiguous tracks is/are unreadable you may have a scratched or dirty diskette. In the

case of a scratched or dirty diskette you should try to copy the readable sectors to another diskette and throw away the damaged diskette.

COPY DISK SECTORS

When first entering this section of Zapsit you will be asked for the SOURCE INFORMATION (drive, density, starting track, starting sector) and the DESTINATION INFORMATION (drive, density, starting track, starting sector). You will then be asked for the SECTOR COUNT. The sector count is the total number of contiguous sectors that you want to copy (1 to 1440 decimal). Enter the number of sectors that you want copied and press ENTER. There is no default for sector count.

If you want to copy an entire disk or from a specific track/sector to the end of the disk, enter a number that is equal to or greater than the number of sectors on the disk. Zapsit will stop when it reaches the end of the count or when it reaches the highest track number + 1, whichever comes first.

Thirty five track double density disks have 630 sectors.
Fourty track double density disks have 720 sectors.
Eighty track double density disks have 1440 sectors.
Thirty five track single density disks have 350 sectors.
Fourty track single density disks have 400 sectors.
Eighty track single density disks have 800 sectors.

The above sector counts assume that the diskettes are in the standard format of 18 sectors/track double density - 10 sectors/track single density. For any non-standard formats you'll have to break out your calculator.

Copy disk sectors allows you to:

Copy sectors (SINGLE or DOUBLE DENSITY) from a disk to different sectors on the SAME disk. If part of the disk is formatted double density and part is formatted single density a mixed single disk copy is allowed.

Copy sectors from a DOUBLE DENSITY disk to a different DOUBLE DENSITY disk.

Copy sectors from a SINGLE DENSITY disk to a different SINGLE DENSITY disk.

Copy sectors from a SINGLE DENSITY disk to a DOUBLE DENSITY disk.

Copy sectors from a DOUBLE DENSITY disk to a SINGLE DENSITY disk.

All copying to a DIFFERENT disk must be done on a TWO DRIVE system.

All sector copying honors the Track/Sector limits.

If the Sectors to be written to fall within the directory track, be sure to set the directory track to the correct track number (menu selection # 3).

If an error occurs during copying, Zapsit will stop and tell you which drive, track, and sector the error occured on and ask you if you want to abort or try again. There is no provision for skipping the bad sector and continuing the copy. You can manualy restart the copy by having it start again from the sector following the bad sector.

ZERO DISK SECTORS

Zero disk sectors allows you to write a value of your choice to the sectors specified. When you first enter this part of Zapsit you will be asked if you really want to do this. If you answer N you will be returned to the main menu. If you answer Y you will be asked for which drive, what density, starting track, starting sector, sector count, and fill byte to use.

The sector count is the number of contiguous sectors that you want to fill (1 to 1440 decimal). There is no default value for the sector count. Pressing ENTER for this question will abort the operation.

The fill byte is the number written to all 256 bytes in the sector and can be any number between 0 and 255 inclusive. The default value for the fill byte is 0.

READ A TRACK

Reading a track allows you to read an entire track into memory, with all of the address marks and information that you don't normally see with a sector read. The output can be to either the screen or printer.

If you choose to have the output go to the screen, the track of information is read into memory with the display set to the first page of information. Then the Display/Modify Memory routines are invoked. You can use all of the commands available to the Display/Modify Memory section. The format of the display is the same as in the Display/Modify Memory section.

Having the output go to the printer allows you to look at all of the information at one time rather than 256 bytes at a time. The entire track will be printed with a blank line between each 256 byte block of information. Doing this gives you a copy of the information in all of the sectors of that track in case

you have to type in a sector that normally couldn't be saved with FORMAT WITHOUT ERASE.

All of the information in the bad sector may not be valid but at least you'll have the opportunity to examine it and decide if you want to type it into a restored sector.

Before you will be able to use the TRACK READ effectively you'll have to know a little about how the disk controller organizes data on a diskette. The following is an example of what a 256 byte block of data would look like if you were to read a track and have it displayed to the screen or printed on the printer. All references to addresses will be taken from column 1.

1	2	3
008F00	D090 9090 9090 9090 9090 9090 9090 9090	....
008F10	9090 9090 9090 9090 9090 9090 9090 9090	....
008F20	9090 9090 9090 9090 9090 9090 9090 9090	....
008F30	9090 9090 9090 9090 9090 9090 9090 9090	....
008F40	FFFF FFFF FFFF FFFF C2A1 A1FE 1500 0101	....
008F50	5DEE 4E4E 4E4E 4E4E 4E4E 4E4E 4E4E 4E4E	).NNNNNNNNNNNNNN
008F60	4E4E 4E4E 4E4E 4E4E 0000 0000 0000 0000	NNNNNNNN........
008F70	0000 0000 14A1 A1FB E5E5 E5E5 E5E5 E5E5	....
008F80	E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5	....
008F90	E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5	....
008FA0	E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5	....
008FB0	E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5	....
008FC0	E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5	....
008FD0	E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5	....
008FE0	E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5	....
008FF0	E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5 E5E5	....

The addresses in column 1 are the addresses in memory where the track of information is now stored. These addresses have nothing to do with the disk storage and are just reference points for finding data in the display or printout.

The hexadecimal information in column 2 is the actual information stored on the disk and is how the disk controller sees it.

The information in column 3 is the ASCII represention of the data in column 2.

In the example above, track 15H of a TRSDOS double density formatted diskette was used. A formatted diskette was used to make it easier to separate the data that you would normally get in a SECTOR READ from the data that only the controller chip usually sees. All of the E5s are data written to the sectors during FORMAT and take the place of the user data that would

normally be in the sector.

The data from address 8F00H thru 8F47H is miscellaneous data leading up to the track & sector information and is generaly called a post index gap. It is the filler that is written on the disk between the INDEX HOLE and the start of the first sector on the track. This first gap is of variable length (between 50 and 80 bytes long). The purpose of the gap is to allow the controller time to get set after it detects the INDEX HOLE. There are also gaps between sectors (22 to 54 bytes long) called sector gaps.

Starting at address 8F48H are 3 bytes (C2 A1 A1) that tell the controller that valid data follows.

The fourth byte (FE) is the ID ADDRESS MARK and tells the controller that the following 6 bytes are track & sector information.

The first of the six bytes is the track # (15H).

The second is the side # (00 = side A, 01 = side B).

The third is the sector # (01).

Fourth is the sector length (01 = 256 bytes).

The fifth and sixth bytes are CHECKSUM numbers that the controller uses to verify that the track & sector ID information is correct.

The next 22 bytes are another gap that gives the controller time to digest the track & sector information. The 22 byte count for the gap is nominal - that is the gap may be 1 or 2 bytes longer or shorter - but is normally 22 bytes long. That gap must be followed by 12 bytes of zeros. That gives a total second gap length of 34 bytes.

The next 3 bytes (14 A1 A1), starting at address 87F4H, tell the controller that valid data follows and the fourth byte is the DATA ADDRESS MARK, in this instance an FBH. Two different DATA ADDRESS MARKS are available to the controller used in the MODEL III. The FBH is called a STANDARD data mark and an F8H is called a DELETED data mark. The type of mark that is written under what circumstances is decided by the programmer who writes the Disk Operating System. So far, it would appear that no standard has been decided upon because each DOS written for the MODEL III handles the DATA ADDRESS MARKS a little bit differently. Generaly, though, they are used to identify the directory sectors from data sectors.

The 256 bytes that follow the DATA ADDRESS MARK are the user data, in this instance they are E5s. The E5H is the FILL BYTE used by TRSDOS during format.

Following the 256 bytes are 2 CHECKSUM bytes (not shown in the example) that the controller uses to verify the validity of the data.

If you think of the above example as one block of information that goes with each sector of data you should not have too much trouble picking your way through a listing of an entire track. Each track is made up of 18 blocks (double density) or 10 blocks (single density) of information.

If you dump a TRACK READ to the printer you will notice that the sector numbers do not go in order. eg. 1,2,3...18. They generaly are staggered so that the diskette does not have to make a complete revolution, during multiple sector reads and writes, to find the next higher sector number. If you take a listing of a TRSDOS track and compare it to a listing of a DOSPLUS track you'll notice that the sectors are not staggered the same. Just one of the reasons why DOSPLUS is faster during disk I/O than TRSDOS.

Here's a hint on how to salvage an important diskette that has bad sectors on it, with a minimal loss of data.

1. Do a disk verify with the bad sectors being reported to the printer. If you don't have a printer you can copy them from the screen. Be sure to include the track numbers of the bad sectors.

2. Dump all of the tracks that have bad sectors to the printer (using READ TRACK) and examine the bad sectors on the listing to see if any or all of the data is there. Sometimes when a sector is lost only the ID information is lost and not the data itself - or only part of the data gets garbled. Again, if you don't have a printer you'll have to copy the information from the screen.

3. Do a FORMAT WITHOUT ERASE on the bad tracks.

4. Type in the data from the listing to the reformatted bad sectors. Do not include the gaps or Track/Sector information.

5. After restoring the data don't forget to BACKUP THE DISK.

Some of the data in the restored sectors may not be valid but it is easier to fix a few bad bytes of data than to try to redo a whole disk full of data.

RETURN TO DOS

To return to the disk operating system, place a system diskette in drive 0, type 9, and press ENTER. You can also use this to switch back and forth between the two versions of zapsit. Just leave the ZAPSIT diskette in drive 0 when selecting 9. You will then be asked which version you want to use.